UseSkill: uma ferramenta de apoio à avaliação de usabilidade de sistemas Web

Matheus M. C. Souza[1], Pedro A. Oliveira[1], Rafael F. Ribeiro[1], Ricardo S. Britto[2], Pedro Santos Neto[1]

[1] Departamento de Ciência da Computação - Universidade Federal do Piauí (UFPI) - Teresina, PI - Brasil

[2]Department of Software Engineering - Blekinge Institute of Technology - Karlskrona - Sweden

{matheusmmcs, petrus.cc, raffael404}@gmail.com, ricardo.britto@bth.se,

pasn@ufpi.edu.br

Abstract. *The usability evaluation of Web systems is crucial to increase end-users' acceptance. However, the complexity and cost associated to usability evaluation discourage the execution of this activity. In this paper we present a tool called UseSkill, which aims to ease the execution of usability evaluation. The tool compares actions performed by "experienced" and "beginners" users. An experiment was performed to evaluate the designed tool. Results show that it is possible to identify usability problems based on the differences between the behavior of the users, in a distributed way and facilitating the use of this kind of testing.*

Resumo. *A avaliação de usabilidade de sistemas Web é crucial para aumentar sua aceitação perante seus usuários finais. Entretanto, a complexidade e os custos associados à avaliação de usabilidade desencorajam a execução dessa atividade. Neste trabalho, apresenta-se a ferramenta UseSkill, que visa facilitar a execução desse tipo de avaliação. A ferramenta compara as ações realizadas por usuários "experientes" e usuários sem conhecimentos no sistema, denominado "novatos". Um experimento foi realizado para avaliar a ferramenta proposta. Os resultados mostram que é possível identificar problemas de usabilidade com base nas diferenças entre o comportamento de usuários, de forma distribuída, facilitando assim a aplicação desse tipo de teste.*

1. Introdução

Aplicações Web fazem parte de diversas atividades cotidianas, como ler notícias, buscar informações, estudar e realizar interações. Devido ao grande número de diferentes aplicações Web disponíveis, a facilidade de uso pode definir o sucesso ou fracasso de uma aplicação [Fernandez et al. 2011, Chen et al. 2012].

Avaliar a usabilidade de aplicações Web é uma atividade muitas vezes negligenciada, devido à complexidade e custo relacionados a tal atividade. Existem diversos métodos para a realização de avaliação de usabilidade de aplicações Web [Fernandez et al. 2011], tais como inspeção [Conte et al. 2009], teste laboratorial [Dumas and Redish 1999] e teste remoto [Castillo et al. 1998]. A seleção de métodos

para avaliar a usabilidade de *softwares* é influenciada pelo tempo, custo, eficiência, eficácia e facilidade de aplicação destes métodos [Ssemugabi and De Villiers 2007].

Muitas atividades relacionadas aos métodos de avaliação de usabilidade podem ser automatizadas, de modo a reduzir o custo e a complexidade relacionada à realização desse tipo de avaliação, liberando os especialistas de tarefas repetitivas, tais como análises manuais de *logs* [Paganelli and Paternò 2002].

Este trabalho propõe uma ferramenta para realizar avaliações de usabilidade remotamente, intitulada UseSkill. A ferramenta proposta captura *logs* durante as avaliações e os analisa de forma automática, baseando-se na diferença de experiência dos usuários, gerando relatórios que devem ser interpretados por especialistas para verificar se o problema inferido realmente existe ou se é um falso positivo.

A ferramenta proposta não sugere a erradicação de testes laboratoriais, mas oferece uma alternativa descomplicada e de baixo custo para avaliar a usabilidade de sistemas Web. Suas principais contribuições podem ser resumidas em:

- **Auxílio a usuários durante avaliações**: a ferramenta possui diversos mecanismos de interação com o participante do teste, disponibilizando um local para registro de impressões e apoio à execução do teste de forma descomplicada;
- **Visualização dos resultados de forma simples e aprofundada**: a ferramenta disponibiliza grafos contendo os caminhos percorridos, com fácil legibilidade, além de métricas de usabilidade e listas com as ações rotuladas, permitindo análises aprofundadas;
- **Não intrusiva**: a UseSkill não exige nenhum tipo de modificação do sistema em teste, mesmo sendo uma ferramenta baseada na coleta de *logs* do lado do cliente.

Este trabalho está estruturado da seguinte forma: na Seção 2 são descritos alguns conceitos fundamentais para uma boa compreensão do trabalho; na Seção 3 é descrita a abordagem proposta, comentando sobre suas características e seus diferenciais; na Seção 4 a ferramenta UseSkill é detalhada, apresentando como utilizá-la e detalhes sobre o funcionamento das etapas de criação e participação de testes, além da geração e análise de relatórios; na Seção 5 é descrito o estudo experimental realizado; na Seção 6 são apresentados os trabalhos relacionados; por fim, na Seção 7, são apresentadas as conclusões e direções para trabalhos futuros.

2. Referencial Teórico

No campo da Engenharia de *Software* (ES), a definição mais aceita de usabilidade foi proposta na norma ISO 9126-1: “usabilidade é a capacidade de um *software* ser compreendido, aprendido, usado, atraente para seu utilizador, e estar em conformidade com as normas/orientações, quando utilizado sob condições especificadas” [ISO 2000]. Na área de Interface Humano-Computador (IHC), o conceito de usabilidade mais amplamente aceito é o da norma ISO 9241-11: “a capacidade de um produto ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” [ISO 1998]. A definição dada pela norma ISO 9241-11 é a que mais se aproxima da perspectiva da interação humana [Fernandez et al. 2011], sendo assim a definição de usabilidade adotada por este trabalho.

De acordo com Dix [Dix 2009], métodos de avaliação de usabilidade podem ser divididos em duas categorias: inspeções, que são baseadas em análises de especialistas em usabilidade ou *designers*; e métodos empíricos, que envolvem a participação de usuários finais do *software*. Segundo Ivory e Hearst [Ivory and Hearst 2001], métodos de avaliação de usabilidade podem ser categorizados sobre quatro aspectos principais: classe de método, tipo de método, tipo de automação e nível de esforço.

Na classe de testes de usabilidade, eles podem ser remotos ou presenciais. A principal diferença é a separação espacial entre os especialistas e os usuários. Dentre os testes remotos, eles podem ser síncronos ou assíncronos. Nesse caso, a separação temporal entre os especialistas e os usuários é a principal diferença entre tais testes [Bruun et al. 2009].

Profissionais de IHC (Interação Humano-Computador) categorizam as ferramentas para avaliação de usabilidade remotas em dois grupos principais: as baseadas em código fonte de páginas Web (conteúdo ou estrutura) e as que analisam dados de utilização (*logs*). Esses *logs* podem ser capturados tanto em servidores (lado do servidor de aplicação), como em navegadores (lado do cliente) [de Santana and Baranauskas 2015].

Capturar *logs* em servidores é tecnicamente mais simples, mas os dados capturados revelam apenas informações relacionadas às páginas que o usuário visitou. Por outro lado, capturar *logs* nos navegadores é computacionalmente mais complexo, mas as informações capturadas são mais detalhadas, apontando, além da página, o elemento e a ação associada.

3. Abordagem Proposta

A ideia central por trás da ferramenta proposta é comparar as ações realizadas por dois tipos diferentes de usuários: "experientes" e "novatos". Os usuários do grupo de experientes (GE) conhecem e utilizam o sistema sob avaliação da maneira esperada. Os usuários do grupos de novatos (GN) nunca utilizaram ou usam com baixa frequência as funcionalidades a serem avaliadas, apesar de conhecerem os processos e conceitos associados. Esta proposta é baseada nos estudos de Uehling e Wolf [Uehling and Wolf 1995], que comparam apenas um experiente e um novato por vez.

Diferentemente de testes laboratoriais, que avaliam um usuário por vez, a captura das ações na abordagem proposta ocorre de forma remota e assíncrona, o que dispensa a presença de especialistas no local e no instante da avaliação. Essa característica também permite que diversos usuários sejam avaliados simultaneamente. A captura das ações ocorre por meio de *web browsers*. Todas as características descritas visam tornar as avaliações menos custosas, mais simples e frequentes durante o ciclo de vida de sistemas Web. A Figura 1 apresenta a ideia da abordagem.

A captura de ações é personalizável, permitindo capturar informações específicas que podem ser úteis durante a análise dos *logs*. Dentre essas informações, algumas são cruciais para diferenciá-las, permitindo realizar comparações entre os fluxos percorridos:

- Tempo: horário que a ação ocorre;
- Tipo da ação realizada: clique, preenchimento de campos, *mouseover*, etc.;
- Elemento que sofreu a ação: botão, *link*, campo de texto, etc.;
- Onde a ação ocorreu: qual página, geralmente utiliza-se a URL.

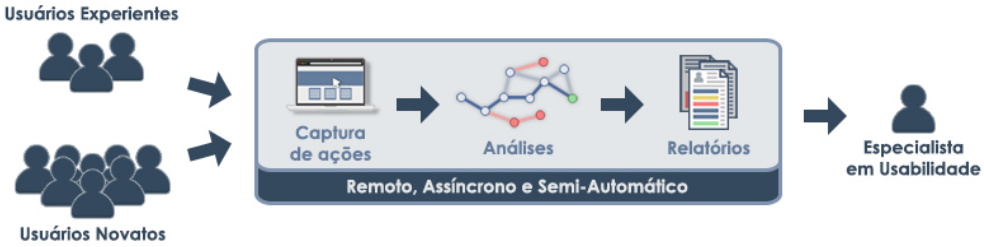

Figura 1. Abordagem proposta para avaliar a usabilidade de sistemas Web.

A análise dos *logs* capturados possui 5 etapas. Inicialmente é necessário criar um grafo contendo todas as ações realizadas por usuários do GE e identificar quais são os caminhos percorridos, quais ações todos os experientes realizaram e qual caminho possui menos interações. A segunda etapa é classificar as ações dos experientes: as que todos os experientes realizaram são denominadas de "ações obrigatórias" (AO); o caminho mais curto define quais são as "ações do caminho ótimo" (ACO); e as demais ações realizadas por usuários experientes são classificadas como "ações normais" (AN).

A terceira etapa da análise é gerar outro grafo contendo as ações realizadas pelo GN. Na etapa 4 essas ações são comparadas com as realizadas por usuários do GE. Caso sejam iguais devem receber a mesma classificação (AN, AO ou ACO), porém caso não se encaixem em nenhuma das classes de ações de experientes, elas são marcadas como "ações desconhecidas" (AD). As ações redundantes mantém suas respectivas classificações, mas são destacadas para que durante a análise sejam identificadas facilmente.

Por fim, na quinta etapa são calculadas algumas métricas sobre as execuções (sessões). Avaliações remotas não possuem controle sobre o foco dos usuário, então a abordagem pouco utiliza métricas com o tempo, focando em métricas relacionadas à quantidade de ações realizadas. As métricas globais, que levam em consideração todas as sessões realizadas, são:

- Média da quantidade de ações realizadas e do tempo despendido por sessão:

$$\overline{x} = \frac{x_1 + x_2 + ... + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Desvio padrão da quantidade de ações realizadas e do tempo despendido por sessão:

$$d_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

Sendo x igual à quantidade de ações ou tempo despendido em cada sessão, n o total de sessões e d_x o desvio padrão da quantidade de ações ou tempo despendido em cada sessão. Com as médias e desvios padrões de ações/tempos é possível calcular o grau de semelhança entre as utilizações de um mesmo grupo e de grupos distintos. As métricas individualizadas, que analisam cada sessão isoladamente, são:

- Percentual de ações em relação ao caminho ótimo (PACO): $p_{aco} = \frac{s}{m}$;
- Percentual de "ações obrigatórias" (PAO): $p_{ao} = \frac{s_{ao}}{t_{ao}}$;
- Percentual de ações contidas no GE (PAGE): $p_{age} = \frac{s_{ge}}{s}$;
- Percentual de ações redundantes ou desconhecidas (PARD): $p_{ard} = \frac{s_{ard}}{s}$.

Sendo m a quantidade de ações no caminho ótimo; s a quantidade de ações na sessão; t_{ao} a quantidade de ações obrigatórias da tarefa; s_{ao} as ações obrigatórias presentes na sessão; s_{ge} as ações AN, AO ou ACO presentes na sessão; s_{ard} as ações redundantes ou desconhecidas presentes na sessão.

As métricas individuais se referem à qualidade das ações realizadas e à eficácia e eficiência de cada sessão, comparando o fluxo percorrido com o caminho ótimo da tarefa, com as ações obrigatórias e com o grupo de usuários experientes.

Para utilizar a abordagem é necessário classificar os usuários como experientes ou novatos e capturar *logs* de utilização com dados sobre o tempo, o tipo de ação realizada, o elemento e a página. Como resultado, são gerados grafos, as ações são classificas e métricas sobre as sessões são calculadas, servindo de indícios para especialistas em usabilidade detectarem possíveis problemas.

4. A Ferramenta UseSkill

Com base na abordagem proposta, foi desenvolvida a ferramenta UseSkill. Ela apoia a realização de testes de usabilidade remotamente em sistemas Web de forma não intrusiva, sem exigir alterações manuais no sistema a ser testado. Concebida em uma plataforma de computação em nuvem, a ferramenta pode ser utilizada por diversos usuários simultaneamente.

A ferramenta é incorporada no ambiente de teste por meio de um *plug-in*, que atualmente está disponível apenas para o *web browser* Chrome (Figura 2). O *plug-in* insere um *script* de captura de ações, sem interferir no funcionamento do sistema sob teste, e apresenta as tarefas que devem ser executadas pelos usuários (com seus respectivos roteiros).

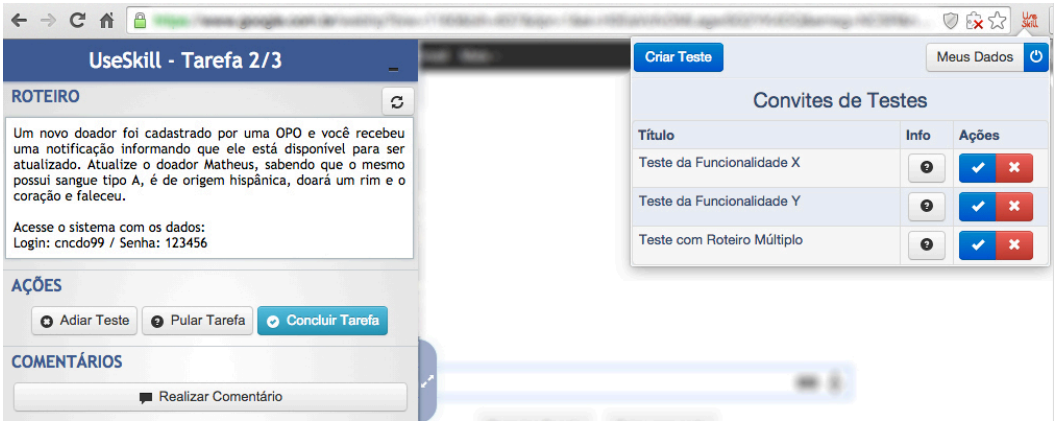

Figura 2. *Plug-in* da UseSkill para o navegador Chrome em ação.

4.1. Funcionamento da UseSkill

O processo de uso da ferramenta é composto por 4 atividades (Figura 3): criar de testes, participar de teste, gerar relatórios e analisar dos relatórios gerados.

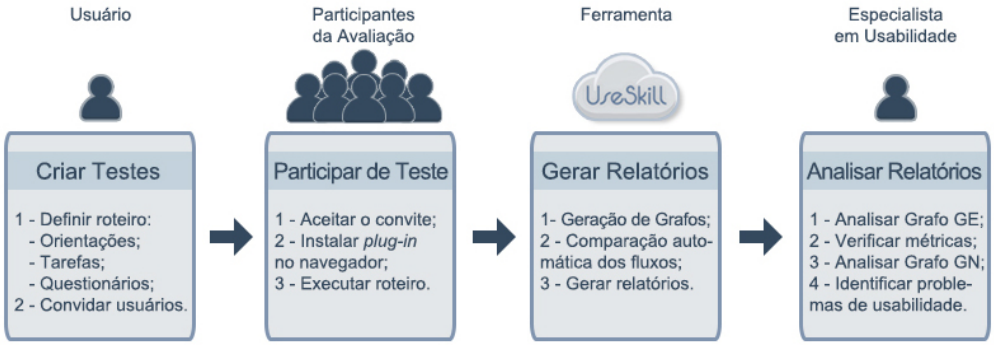

Figura 3. Processo de uso da UseSkill.

4.1.1. Criar Teste

A UseSkill permite a realização de avaliações de usabilidade controladas, onde cada avaliação possui um texto para contextualização dos participantes, uma lista de tarefas e perguntas, além de um conjunto de usuários experientes e novatos. Cada tarefa possui um título, um roteiro detalhando o que deve ser realizado, além do endereço no sistema Web para execução da tarefa.

A ferramenta também permite definir perguntas antes e/ou depois de cada tarefa, definindo um *roadmap* de execução. Cada pergunta possui um título, um texto para a pergunta e, caso a resposta seja objetiva, as alternativas. A Figura 4 apresenta a tela da UseSkill responsável pela etapa de criação de um teste.

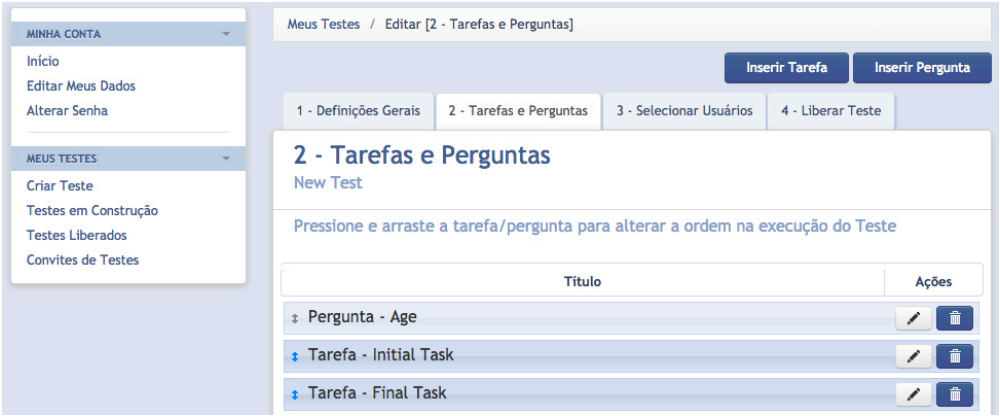

Figura 4. Criação de um teste de usabilidade na UseSkill.

Após definir as tarefas e questionários da avaliação, é necessário convidar os usuários participantes. Ao enviar o convite, o usuário deve ser classificado como experiente ou novato. Em seguida, os usuários recebem um email de notificação para que iniciem a avaliação enquanto a UseSkill captura seus dados de navegação.

4.1.2. Participar de Teste

Cada usuário convidado pode aceitar ou recusar a participação em um teste. Caso o usuário aceite o convite, a ferramenta irá capturar *logs* de utilização durante a avaliação. Por padrão, os seguintes dados são capturados:

- Dados gerenciais, relacionadas ao contexto de utilização:
  - Informações sobre o sistema operacional utilizado;
  - Dimensões da tela;
  - Versão do navegador;
- Dados de utilização que serão comparados:
  - Tempo: horário em que cada ação ocorreu;
  - Tipo de ação: por padrão pode ser clique, *mouseover*, preenchimento de campo, carregamento de páginas, botão voltar e atualizar página;
  - Elemento que sofreu a ação: identificado pelo *XPath* do elemento;
  - Onde a ação ocorreu: a partir da URL da página.

Para realizar a captura dos dados citados, a ferramenta insere um *script*, na linguagem Javascript, que captura eventos na interface e os envia para o servidor da UseSkill ao final de cada tarefa. A Figura 2 apresenta o *plug-in* em execução.

4.1.3. Geração de Relatórios

Essa é a etapa responsável por prover relatórios que possam auxiliar especialistas em usabilidade na identificação de problemas. Os relatórios são individualizadas e gerados a partir dos fluxos de ações capturados em cada uma das tarefas do teste. A Figura 5 apresenta o grafo de um relatório gerado pela UseSkill.

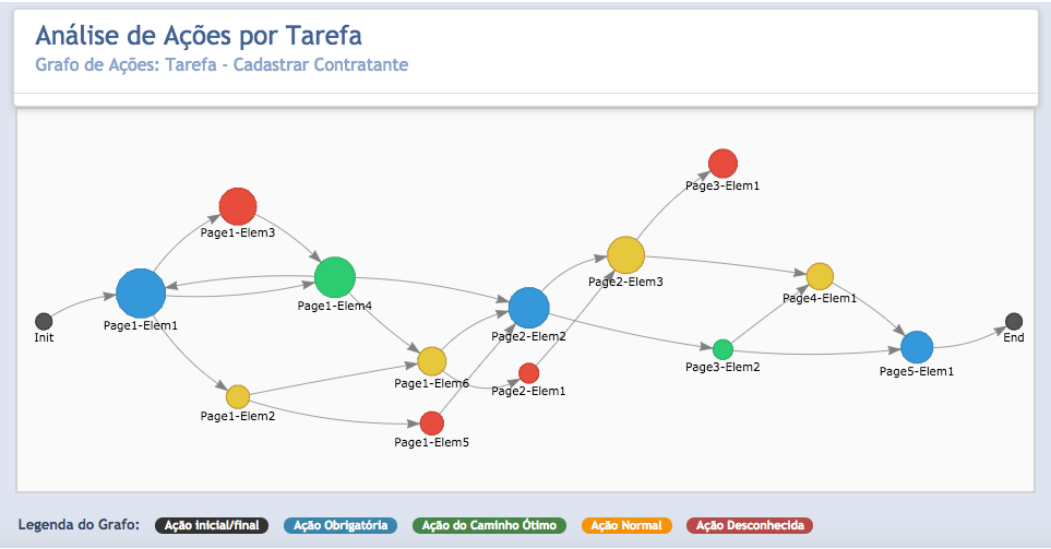

Figura 5. Grafo gerado pela UseSkill contendo as ações realizadas por usuários do grupo de novatos (GN) em determinada tarefa.

Primeiramente, a ferramenta gera uma grafo concatenando as execuções dos usuários do "grupo de experientes" (GE) e classificando suas ações como "ação obrigatória" (AO), "ação do caminho ótimo" (ACO) ou "ação normal" (AN). Em seguida é gerado outro grafo contendo as execuções dos usuários do "grupo de novatos" (GN), sendo que cada ação é comparada e classificada a partir das ações semelhantes realizadas no GE. Caso a ação não pertença ao GE, ela é rotulada como "ação desconhecida" (AD).

Os vértices dos grafos representam as ações e as arestas correspondem à sequência entre as ações. A UseSkill propõe uma visualização intuitiva e personalizável. Cada vértice possui tamanhos distintos, que variam de acordo com a quantidade de vezes que a ação foi realizada, e sua cor indica a classificação da ação, facilitando a legibilidade e interpretação dos dados.

As ações são coloridas de acordo com suas classificações: as "normais" são amarelas, "obrigatórias" são azuis, "caminho ótimo" são verdes e "desconhecidas" são vermelhas. Caso haja muitas ações no grafo, é possível filtrá-las de acordo com o tipo de ação e quantidade de vezes que ela foi realizada. O Algoritmo 1 apresenta o pseudocódigo utilizado para construir os grafos de utilização.

Algorithm 1 Algoritmo para construção dos Grafos de ações realizadas

Entrada: Coleção de fluxos de ações ordenadas pelo tempo.
Saída: Grafos de ações do GE e GN.

```
1:  GGE = novo Grafo; GGN = novo Grafo;                    ▷ Grafos do GE e GN
2:  AcoesGlobais = nova Lista;                  ▷ Lista contendo todas as ações distintas
3:  para cada fluxo F na Coleção de fluxos faça
4:      pos = 1; AcoesFluxo = nova Lista;
5:      se F.usuario == experiente então
6:          G = GGE;
7:      senão
8:          G = GGN;
9:      fim se
10:     para cada ação A no fluxo F faça
11:         V = novo Vértice; V.acao = A;
12:         se !G.contem(V) então
13:             V.qtd = 1; V.posicao = pos;
14:         senão
15:             VG = G.get(V); VG.qtd = VG.qtd+1;
16:         fim se
17:         se AcoesFluxo.contem(A) então
18:             A.repetida = true;                     ▷ Marcação de ações repetidas
19:         fim se
20:         AcoesFluxo.add(A); AcoesGlobais.add(A); pos++;
21:     fim para
22: fim para
```

O caminho ótimo é encontrado comparando todos os fluxos de usuários "experientes", sendo que o fluxo com a menor quantidade de ações é definido como caminho ótimo (CO) e em caso de fluxos com tamanhos iguais, o critério de desempate é o tempo. Com

os grafos criados, as ações repetidas marcadas e o caminho ótimo encontrado, a próxima etapa é realizar a classificação das ações.

Na etapa de classificação, todas as ações passam por uma sequência de verificações: se a ação fizer parte do caminho ótimo, ela é classificada como ACO; caso contrário, se a ação fizer parte de todos os fluxos dos experientes, ela é classificada como AO; caso contrário, se a ação fizer parte de ao menos um fluxo experiente, ela é classificada como AN; caso contrário, ela é definida como AD.

Para cada realização de tarefa, a ferramenta gera uma tabela de detalhamento de ações, contendo todas as ações realizadas e suas respectivas classificações, além das métricas: percentual de ações em relação ao caminho ótimo (PACO), percentual de "ações obrigatórias" (PAO), percentual de ações contidas no GE (PAGE) e percentual de ações redundantes ou desconhecidas (PARD).

4.1.4. Análise dos Relatórios

Com todos os relatórios gerados, um especialista em usabilidade deve analisá-los para identificar possíveis pontos com problemas de usabilidade.

Primeiramente, o grafo do GE deve ser analisado, verificando se os usuários experientes realizaram a tarefa adequadamente ou se cometeram falhas ou redundâncias. Essa análise inicial além de identificar problemas de usabilidade com base apenas em ações de usuários experientes, é determinante para que sejam gerados bons relatórios, pois se o GE cometer falhas ou redundâncias, os usuários do GN podem cometer os mesmos problemas, porém serão mascarados, pois não haverá diferenças entre experientes e novatos em locais com problemas.

Em seguida, as métricas, em especial a média e desvio padrão da quantidade de ações e tempo, serão usadas como indícios sobre a complexidade da tarefa e sobre a discrepância entre as execuções do mesmo grupo de usuários. As métricas das sessões também permitem avaliar o quão bom foi cada execução. A PACO, PAGE e PAO indicam a eficácia do fluxo em relação aos experientes, sendo a última de suma importância para verificar se o fluxo passou pelos pontos obrigatórios. Apesar das métricas citadas, elas podem ocultar problemas com redundância, que são apontadas pela métrica PARD.

A terceira etapa é analisar o grafo do GN, identificando vértices na cor vermelha com maior raio, indicando quais ações "desconhecidas" foram mais realizadas. A tabela contendo o detalhamento das ações permite uma análise mais aprofundada dos problemas e de suas causas. Além da tabela de detalhamento das ações realizadas pelo usuário, a ferramenta disponibiliza a tabela de completude, que apresentada todas as ações obrigatórias da tarefa, destacando as realizadas pelo usuário. Essa tabela auxilia na identificação de quais ações obrigatórias foram menos realizadas, apontando quais locais em que os usuários mais encontraram dificuldades, além da completude de cada usuário por tarefa.

Complementar aos relatórios gerados é possível ler comentários enviados durante os testes e relacionar os resultados às respostas dos questionários. Com esses dados disponíveis em nuvem é possível avaliar remotamente a usabilidade de sistemas Web.

4.2. Desafios e limitações

Apesar das grandes vantagens providas pela UseSkill é necessário ressaltar suas limitações. A ferramenta realiza avaliações especificamente para sistemas Web, sendo assim sua primeira restrição. Para sanar parte dessa restrição, uma versão UseSkill mobile está em desenvolvimento. A necessidade de possuir usuários experientes e novatos também pode dificultar seu uso, pois nem sempre será fácil encontrar usuários com níveis de experiência distintos disponíveis para avaliar o sistema.

Outra limitação é que a qualidade dos relatórios gerados depende diretamente da qualidade das execuções realizadas por usuários. Se usuários experientes errarem durante a execução das tarefas, eles podem mascaram problemas de usabilidade, pois não haverá diferença entre experientes e novatos em pontos problemáticos. Para amenizar tal problema, usuários experientes ou o responsável pela avaliação podem descartar ou ignorar fluxos do GE que possuírem problemas.

5. Estudo Experimental

Para avaliar os resultados da ferramenta, foi realizado um estudo experimental. O objetivo do experimento foi avaliar os resultados da realização de testes de usabilidade laboratoriais e de avaliações de usabilidade com o apoio da ferramenta UseSkill.

Para o experimento foi utilizado um sistema de gestão de cooperativas médicas com clientes em vários estados do Brasil. Ele possui cerca de 130 funcionalidades e aproximadamente 80 KLOC (*Kilo Lines of Code*). As variáveis independentes são as técnicas para avaliação de usabilidade e as variáveis dependentes são o tempo e a quantidade de problemas identificados. O tempo, registrado em minutos, envolve não apenas a execução das tarefas, mas também o tempo gasto com a avaliação por especialistas. Foram contabilizados todos os problemas de usabilidade identificados pelas técnicas, independente da criticidade do mesmo.

Por dificuldades inerentes à realização do experimento, foi realizado um *quasi-experiment*, pois os participantes foram escolhidos de forma não aleatória e baseada na conveniência, de acordo com a disponibilidade dos participantes [Wohlin et al. 2000]. Dentre os participantes, 29 são alunos de graduação do curso de Ciência da Computação e três desenvolvedores do sistema de gestão de cooperativas médicas utilizado nas avaliações.

5.1. Hipóteses

O experimento observou as seguintes hipóteses nulas e suas hipóteses alternativas correspondentes:

$H_0$1: Não há diferença entre o tempo despendido no teste laboratorial e na UseSkill.
H_A1: O tempo despendido na avaliação da UseSkill é diferente que no teste laboratorial.
$H_0$2: Não há diferença entre o custo com especialistas no teste laboratorial e na avaliação da UseSkill.
H_A2: O custo com especialistas na avaliação da UseSkill é diferente do custo do teste laboratorial.
$H_0$3: Não há diferença entre a quantidade de problemas de usabilidade identificados no teste laboratorial e na avaliação da UseSkill.
H_A3: A quantidade de problemas de usabilidade identificados na avaliação da UseSkill é diferente que no teste laboratorial.

5.2. Desenho Experimental

O experimento foi planejado seguindo o desenho de "um fator com dois tratamentos completamente aleatórios"; os participantes atribuídos aos dois grupos foram selecionados aleatoriamente, embora a seleção dos participantes do experimento tenha sido feita por conveniência. O Grupo 1 possui cinco alunos de graduação que avaliaram o sistema Web usando o método de teste laboratorial. O Grupo 2 possui 27 pessoas que utilizaram a UseSkill para avaliarem o sistema Web (24 alunos, desempenharam o papel de usuários novatos, e três desenvolvedores que desempenharam o papel de usuários experientes).

Tabela 1. Desenho experimental aplicado.

Grupos	Contextualização	Laboratorial	UseSkill
Grupo 1	X	X	
Grupo 2	X		X

Foram selecionados mais participantes para o Grupo 2, pois ao realizar avaliações remotas é possível acrescentar participantes sem aumentar significativamente o custo [Bastien 2010]. Ambos os grupos passaram por um processo de contextualização acerca do sistema de gestão de cooperativas médicas. Adicionalmente, os membros do Grupo 2 também foram apresentados à UseSkill. Foi frisado que é os participantes deveriam apenas avaliar o sistema Web, e não a ferramenta UseSkill. A Tabela 1 apresenta o desenho do experimento.

5.3. Execução e Análise

Cada execução foi analisada, mensurando os tempos gastos e a quantidade de problemas de usabilidade identificados. O levantamento de problemas de usabilidade durante o teste laboratorial se deu a partir de observações anotadas em planilhas por um especialista durante o teste, além de análises dos vídeos gravados e questionários pós teste. Um total de 13 problemas de usabilidade distintos foram identificados durante a execução do teste laboratorial.

Para identificar problemas de usabilidade com a UseSkill, foram utilizados os grafos, listas de ações e métricas. O especialista conseguiu identificar 10 problemas de usabilidade distintos por meio da UseSkill, vide Figura 6.

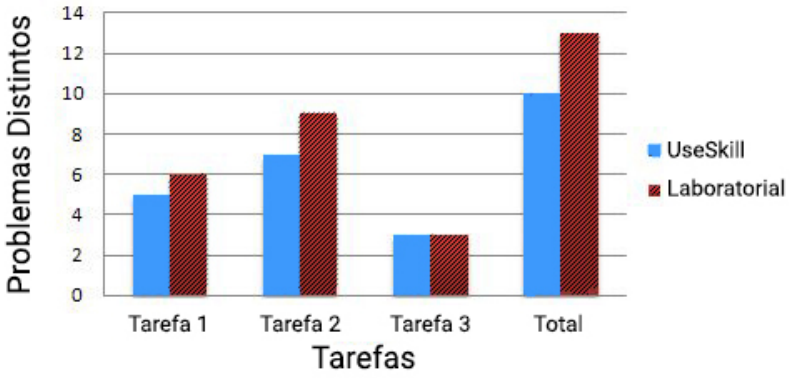

Figura 6. Quantidade de problemas distintos detectados em cada técnica.

Quanto ao tempo gasto em cada um dos métodos, foram comparados os tempos referentes à execução da avaliação e análise dos dados. A etapa de planejamento das duas técnicas foi realizada em conjunto, sendo desconsiderada nesta comparação. Quanto ao tempo para realizar cada tarefa, a média do teste laboratorial foi 7,5 minutos, enquanto na UseSkill a média dos experientes e novatos foi 8,9. Observou-se que os participantes do teste laboratorial são mais focados, realizando as tarefas em um período de tempo menor que os participantes da UseSkill.

Apesar da pequena diferença entre os tempos de execução para cada tarefa, o tempo total de execução do teste laboratorial com cinco participantes foi de 2 horas e 52 minutos. No caso da UseSkill os 24 alunos utilizaram 1 hora e 19 minutos, e os três desenvolvedores levaram 49 minutos, totalizando 2 horas e 8 minutos.

A análise dos dados coletados e dos vídeos do teste laboratorial durou 4 horas e 47 minutos, enquanto a análise das tabelas da UseSkill consumiu 3 horas e 26 minutos. Ao contabilizar todo o tempo consumido com execução e análise, o teste laboratorial necessitou de 7 horas e 39 minutos, enquanto a UseSkill consumiu 5 horas e 34 minutos, uma redução de 27,3% no tempo necessário para a avaliação, vide Tabela 2.

Tabela 2. Duração das avaliações realizadas.

Etapas	Laboratorial	UseSkill
Execução	2 horas e 52 minutos	2 horas e 08 minutos
Análise	4 horas e 47 minutos	3 horas e 26 minutos
Total	7 horas e 39 minutos	5 horas e 34 minutos

5.4. Discussão

O teste laboratorial consumiu no total 7 horas e 39 minutos, enquanto a UseSkill exigiu 5 horas e 34 minutos para sua realização, uma redução de 27,3% no tempo necessário. Apesar de estar rejeitando a hipótese $H_0 1$, este resultado depende diretamente da quantidade de participantes do teste laboratorial e do tamanho do teste.

Quanto ao gasto com especialistas, o teste laboratorial exigiu a presença do especialista em um período total de 7 horas e 39 minutos, pois além da análise dos dados, foi necessária sua presença durante a realização do teste. A UseSkill exigiu 3 horas e 26 minutos do especialista para a análise dos dados, reduzindo em 55% o custo com especialistas. Rejeitando assim a hipótese $H_0 2$.

Em relação à quantidade de problemas detectados, o teste laboratorial detectou 13 problemas distintos, enquanto a UseSkill detectou apenas 10, uma redução de 23,1%. Desta forma a hipótese $H_0 3$ é rejeitada. A ferramenta proposta surge como uma boa opção para médias e pequenas empresas de desenvolvimento de *software*, que por conta do alto custo, não conseguem realizar testes de usabilidade laboratoriais.

As evidências apontadas pelo estudo experimental podem sofrer ameaças à validade. Dentre elas, pode ter havido maturação, mas para evitar o cansaço dos participantes, os treinamentos realizados e as tarefas selecionadas foram curtos e executados em pouco tempo. Vale ressaltar também que não houve mortalidade no experimento; todos os convidados participaram do início ao fim. Também foram evitadas as ameaças sociais;

durante o treinamento foram apresentados os objetivos do estudo e evidenciado que os dados seriam avaliados anonimamente.

A seleção dos participantes foi feita por conveniência, o que limita a capacidade de generalização dos resultados observados. Outro fator destacável é que os participantes não trabalham com gestão de cooperativas médicas e podem ter ficados confusos com alguns conceitos, impactando nos resultados.

6. Trabalhos Relacionados

Avaliação de usabilidade remota automática ou semiautomática é um importante instrumento para apoiar o desenvolvimento de aplicações Web modernas. A automatização desse tipo de avaliação reduz o seu custo, pois o tempo necessário para avaliar a usabilidade diminui significativamente e a necessidade de avaliadores também é reduzida ou mesmo eliminada [Ivory and Hearst 2001].

A ferramenta *User Action Graphing Effort* (UsAGE) [Uehling and Wolf 1995] é um dos trabalhos pioneiros na automatização de avaliações de usabilidade. A ferramenta é baseada na comparação de um usuário "experiente" e um "novato", identificando pontos de diferença nas suas interações. Com essa ferramenta é possível avaliar *softwares desktop* que possuem interfaces construídas a partir da ferramenta TAE Plus, que dentre outras funcionalidades, possibilita a gravação automática das ações realizadas na interface. A UsAGE deve ser utilizada durante sessões de testes laboratoriais, sendo que e a partir dos *logs* capturados são realizadas comparações automáticas que geram um grafo contendo as ações realizadas por cada usuário. A UseSkill baseou-se na UsAGE, entretanto com foco em avaliações de aplicações Web e comparações de grupos de usuários "experientes" e "novatos". Outro diferencial é a classificação das ações, tornando mais simples a interpretação dos resultados.

Outra ferramenta que utiliza um método semelhante ao proposto é a USABILICS, que realiza avaliações remotas e semiautomáticas de usabilidade de aplicações Web. Ao criar cada tarefa do teste com a ferramenta, são definidas as ações esperadas. Em seguida, a ferramenta compara as ações esperadas com as ações realizadas por usuários durante os testes, calculando a similaridade entre essas sequências de eventos. Para capturar os eventos dos usuários, são necessárias alterações no código fonte da aplicação a ser testada. Como resultado, a ferramenta calcula o índice de usabilidade de cada tarefa [de Vasconcelos and Baldochi 2012]. Apesar das semelhanças, a UseSkill gera tabelas e grafos comparando as ações realizadas por "novatos" e "experientes", facilitando a identificação de possíveis problemas de usabilidade, além de não ser intrusiva.

A WELFIT [de Santana and Baranauskas 2015] é uma ferramenta que suporta testes remotos/não-remotos, síncronos/assíncronos, e formais/informais. Ela realiza a captura de *logs* automaticamente do lado do cliente, exigindo alteração do código fonte da aplicação. Durante a comparação automática dos *logs*, a ferramenta leva em consideração a distância, a partir da heurística *Sequence Alignment Method* (SAM) e o tempo médio de cada evento. Os resultados obtidos a partir da ferramenta são em forma de relatórios estatísticos e grafos dos eventos capturados.

Apesar da abordagem da WELFIT ser semelhante à proposta neste trabalho, ela necessita de alterações no código fonte da aplicação, além de não permitir personalizar quais eventos devem ser capturados. Outro possível problema é que caso haja uma

grande massa de *logs*, a legibilidade dos *grafos* gerados por ela fica comprometida, dificultando a identificação pontual dos problemas de usabilidade. Para amenizar esse problema, a UseSkill permite a personalização dos eventos capturados e a configuração da visualização de grafos, além de apresentar tabelas contendo os *logs* classificados e detalhados.

7. Conclusão

Este trabalho apresentou a UseSkill, uma ferramenta que apoia a avaliação de usabilidade de sistemas Web remotamente de forma assíncrona e que não necessita de alterações no sistema a ser testado. Ela baseia-se na divisão dos usuários em "experientes" e "novatos", comparando as ações realizadas por eles automaticamente e indicando possíveis problemas de usabilidade. Para cada participante da avaliação é necessária apenas a instalação de um *plug-in* para o navegador Chrome. Essas características facilitam a ampla utilização da ferramenta e corroboram com o intuito de reduzir os custos e a complexidade de avaliações de usabilidade realizadas por usuários reais.

A importância e a necessidade de avaliar a usabilidade de sistemas Web de forma simples e a baixo custo motivou a realização deste trabalho. A ferramenta proposta utiliza apenas *logs* capturados durante as sessões para identificar possíveis pontos problemáticos, não necessitando da definição de modelos de utilização. A UseSkill auxilia os participantes durante as sessões do teste, abrindo as tarefas e questionários na ordem definida no teste e possibilitando ao usuário realizar comentários durante a avaliação. Os relatórios da ferramenta são completos e de fácil interpretação, contendo grafos, tabelas de ações classificadas e métricas. Essa gama de relatórios permite que especialistas avaliem remotamente e tenham acesso a detalhes das execuções, caso necessário.

O estudo experimental realizado demonstrou que a ferramenta pode reduzir o custo e a complexidade da realização de avaliações de usabilidade. Apesar da ferramenta ter detectado menos problemas de usabilidade que a abordagem laboratorial, a UseSkill detectou todos os problemas impeditivos, que influenciaram diretamente na eficácia e eficiência da utilização. Os bons resultados apresentados pela ferramenta também motivam trabalhos futuros, como a extensão da técnica para dispositivos móveis que está em andamento.

8. Agradecimentos

Os autores agradecem aos participantes do estudo experimental realizado e à empresa Infoway Ltda pelo incentivo e importantes contribuições para a pesquisa.

Referências

[Bastien 2010] Bastien, J. C. (2010). Usability testing: a review of some methodological and technical aspects of the method. *International Journal of Medical Informatics*, 79(4):e18–e23.

[Bruun et al. 2009] Bruun, A., Gull, P., Hofmeister, L., and Stage, J. (2009). Let your users do the testing: a comparison of three remote asynchronous usability testing methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1619–1628. ACM.

[Castillo et al. 1998] Castillo, J. C., Hartson, H. R., and Hix, D. (1998). Remote usability evaluation: can users report their own critical incidents? In *CHI 98 Cconference Summary on Human Factors in Computing Systems*, pages 253–254. ACM.

[Chen et al. 2012] Chen, S.-C., Yen, D. C., and Hwang, M. I. (2012). Factors influencing the continuance intention to the usage of web 2.0: An empirical study. *Computers in Human Behavior*, 28(3):933 – 941.

[Conte et al. 2009] Conte, T., Massolar, J., Mendes, E., and Travassos, G. H. (2009). Web usability inspection technique based on design perspectives. *Simpósio Brasileiro de Engenharia de Software (SBES)*, 1.

[de Santana and Baranauskas 2015] de Santana, V. F. and Baranauskas, M. C. C. (2015). Welfit: A remote evaluation tool for identifying web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, 76:40–49.

[de Vasconcelos and Baldochi 2012] de Vasconcelos, L. G. and Baldochi, Jr., L. A. (2012). Towards an automatic evaluation of web applications. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 709–716, New York, NY, USA. ACM.

[Dix 2009] Dix, A. (2009). *Human-computer interaction*. Springer.

[Dumas and Redish 1999] Dumas, J. S. and Redish, J. (1999). *A practical guide to usability testing*. Intellect Books.

[Fernandez et al. 2011] Fernandez, A., Insfran, E., and Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817.

[ISO 1998] ISO (1998). Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: Guidance on usability, ISO 9241-11. *International Organization for Standardization*.

[ISO 2000] ISO (2000). Software engineering - product quality - part 1: Quality model, ISO/IEC 9126-1. *International Organization for Standardization*.

[Ivory and Hearst 2001] Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516.

[Paganelli and Paternò 2002] Paganelli, L. and Paternò, F. (2002). Intelligent analysis of user interactions with web applications. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 111–118. ACM.

[Ssemugabi and De Villiers 2007] Ssemugabi, S. and De Villiers, R. (2007). A comparative study of two usability evaluation methods using a web-based e-learning application. In *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 132–142. ACM.

[Uehling and Wolf 1995] Uehling, D. L. and Wolf, K. (1995). User action graphing effort (usage). In *Conference companion on human factors in computing systems*, pages 290–291. ACM.

[Wohlin et al. 2000] Wohlin, C., Runeson, P., Host, M., Ohlsson, M., Regnell, B., and Wesslen, A. (2000). Experimentation in software engineering: an introduction. 2000.